Homework 2

Due: September 23, 2025 at 11:59pm

Instructions

- See https://rpmml.github.io/homework_instructions/ for instructions on accessing the accompanying code and submitting homework via GradeScope.
- See https://rpmml.github.io/policies for additional policies regarding collaboration, LLMs, and late submissions.
- Use Piazza or attend office hours to ask questions about the homework.

Part 1: Written Exercises

Practice with State Estimation (15 points)

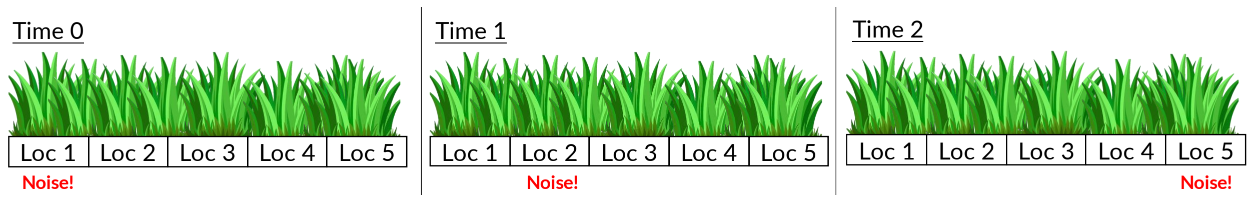

Figure 1: Illustration for state estimation question.

A frog is hopping between 5 locations that are hidden behind tall grass (Figure 1). The frog makes a noise when in each location, but the noise is muffled:

- If the frog is at Location 1, a noise is heard at Location 1 with probability 0.75 and heard at Location 2 with probability 0.25.
- If the frog is at Location 5, a noise is heard at Location 5 with probability 0.75 and heard at Location 4 with probability 0.25.
- If the frog is at Location x, for $1 < x < 5$, a noise is heard at Location x with probability 0.5, Location $x - 1$ with probability 0.25, and Location $x + 1$ with probability 0.25.

After making a noise, the frog may hop to an adjacent location:

- If at Location 1, it hops to Location 2 with probability 0.5 and stays in place otherwise.
- If at Location 5, it hops to Location 4 with probability 0.5 and stays in place otherwise.
- If at Location x, for $1 < x < 5$, it hops to location $x - 1$ with probability 0.25, to location $x + 1$ with probability 0.25, and stays in place otherwise.

You hear a noise at Location 1 at $t = 0$, Location 2 at $t = 1$, and Location 5 at $t = 2$. Calculate and report your belief about the frog's true location after each of the noises are heard. Do filtering only, not smoothing; i.e., when calculating your belief about $t = 0$, consider only the noise at Location 1, not the future noises. Show your work.

Pause to Ponder (No Points & No Submission Required)

Under what circumstances, if any, can you be absolutely certain about the frog's location?

Practice with POMDP Formulation (10 points)

On Monday morning, you see a painting at a local store that is selling for \$100. You have a keen eye for art and you believe there is a small chance this painting is the original work of a famous artist, which would make it worth \$10,000. If the painting is not original, then it is only worth \$10. At this moment, you think there is a 20% chance that the painting is original and an 80% chance that it is not. You also happen to be friends with an art expert who could tell you with absolute certainty whether the painting is original. Unfortunately, your friend is on vacation, and the earliest that they could assist you is first thing Wednesday morning. With each passing day, there is a 10% chance that someone else will buy the painting. Formulate a POMDP that captures this decision-making problem.

Practice with POMDP Expectimax Search (20 points)

Draw a complete expectimax search tree for the POMDP you formulated in the previous question.[1] Make sure that all values are shown in the tree. Annotate the optimal actions at each node. You can hand-draw the figure or use any software to produce it.

Pause to Ponder (No Points & No Submission Required)

Why might losing money on unoriginal painting "feel worse" than gaining money on an original one, even if the amounts lost and gained were the same?

Part 2: Coding Exercises

Policy Trees (10 points)

In this question, you will implement a data structure for a policy tree, and then use that data structure to create three specific policy trees for a Tiger POMDP. There are no requirements for your data structure other than that the unit tests must pass. Examine the unit tests in `test_policy_tree.py` to make sure you understand the question. No written answer required.

[1] There are multiple ways to formulate the POMDP. Feel free to revise your answer to the first question if it makes this second question easier.

Pause to Ponder (No Points & No Submission Required)

How would you change our implementation of expectimax search to extract a policy tree?

Conformant Planning (20 points)

A *conformant plan* is a sequence of actions that is guaranteed to achieve some performance regardless of which initial state and transitions are sampled from a POMDP. For example, if you are in a dark room but you know that there is a light switch somewhere on a wall, then sweeping your hand along the wall is a conformant plan—eventually you will touch the switch, regardless of where you start. In this question, you will implement two conformant planners: a "good" conformant planner that is guaranteed to find a plan that attains total rewards 1.0 or greater; and a "bad" conformant planner that is guaranteed to find a plan that attains total rewards -1.0 or less. Both planners must report if no conformant plan is possible. Examine the unit tests in `test_conformant_planning.py` to make sure you understand the question. Note that your program must return the shortest plan that meets these guarantees. Suggested approach: For these unit tests, it is computationally feasible to explore every possible path. This means for every plan length, enumerating all possible combinations of actions; then for every action plan, explore the tree to find the worst and best case scenarios. We have provided helper functions (that you must implement) which may be useful on the way. No written answer required.

Pause to Ponder (No Points & No Submission Required)

Our approach here was fairly "brute force." Can you think of any tricks to speed this up?

Most Likely State Estimation (10 points)

Implement `MostLikelyStateEstimator` in `most_likely_state.py`. This estimator should update the belief state to be a distribution with just a single state that has 1.0 probability. In addition, implement `CustomPOMDP` in such a way that using this most likely state estimator leads to suboptimal performance. Examine the unit tests in `test_most_likely_state.py` to make sure you understand the question. No written answer required.

Pause to Ponder (No Points & No Submission Required)

Is it ever possible to perform information-gathering when using the most likely state estimator?

Part 3: Your Turn to Teach

Create a Question (10-15 points)

Create a new question of any kind (writing, coding, etc.) that pertains to the material covered in this problem set. To receive full credit, you only need to write the question. To receive up to 5 points of extra credit, you should also provide a solution. Extra credit points will also be awarded based on creativity and effort. Please indicate in your answer (1) whether your question can be used (with modifications) in future versions of this course; and (2) whether you would like to be credited by name if your question is used.

Part 4: Feedback

General Course Feedback (5 points)

How can we improve the course in future years? Any and all feedback is welcome. Please comment on this problem set, lectures, and anything else. Some feedback is required for full credit. If you prefer to additionally submit anonymous feedback, please do so through the course website.